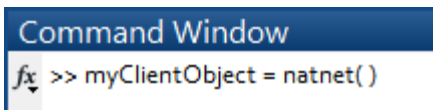


## Guide to using the MATLAB natnet class

### I. Setup

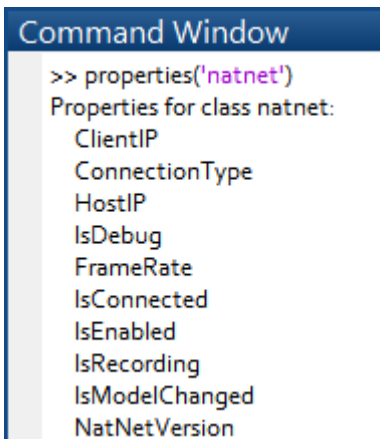
The natnet class is a wrapper for the NatNetML assembly and provides a simplified interface for managing the native members in MATLAB. The class definition and supporting code should be placed on the MATLAB PATH. The implementation automatically disposes running connections when ending a streaming session, along with basic object management. The NatNetML assembly must be loaded into the MATLAB session. This is handled automatically and the first time the class is used the user is prompted to find the NatNetML.dll file in the Windows file browser. A reference to this location is used in future MATLAB sessions.

To create an instance of the class, simply call the class with no input arguments and store it in a variable.



```
Command Window
fx >> myClientObject = natnet( )
```

The available properties to the class can be seen with the following command.



```
Command Window
>> properties('natnet')
Properties for class natnet:
    ClientIP
    ConnectionType
    HostIP
    IsDebug
    FrameRate
    IsConnected
    IsEnabled
    IsRecording
    IsModelChanged
    NatNetVersion
```

And the available methods.

```
Command Window
>> methods('natnet')

Methods for class natnet:

addlistener      getFrame         startPlayback
connect          getFrameRate     startRecord
cycleRecord      getModelDescription stopPlayback
delete           liveMode         stopRecord
disable          natnet           verifyConnection
disconnect       setAssemblyPath
editMode         setPlaybackTakeName
enable          setTakeName

Methods of natnet inherited from handle.
```

Creating an instance of the class does not automatically connect the object to a host application. After enabling the OptiTrack Steaming Engine in Motive or any other Server, change the ConnectionType, ClientIP and HostIP properties to reflect your network setup.

```
Command Window
fx >> myClientObject.HostIP = '123.0.10.1'
    myClientObject.ClientIP = '123.0.10.100'
```

Then enter the following line to connect the natnet object to the host.

```
Command Window
fx >> myClientObject.connect
```

When creating a natnet class instance, the default host and client IP address is '127.0.0.1', which is the local loopback address. The natnet object will fail to connect if the network address of the host or client is incorrect.

## II. Polling

The natnet interface has a method to poll mocap data called getFrame. getFrame returns the data structure of the packet. Polling is supported but not recommended due to accessing errors. The function, poll.m, provides a simple example showing out to poll the frames of mocap data. After connecting a natnet object to the host server, run the polling script to acquire the packets of data in the main workspace.

```
poll.m  x  +
1
2 -  for i = 1:10
3 -      tic
4 -      mdata ( i ) = c.getFrame;
5 -      pause( 0.008333 ) % set to appropriate frame rate, seconds
6 -      toc
7 -  end
8
9 -  display( 'polling stopped' )
10
```

### III. Event Callbacks

The natnet class implements a simple interface to use event callbacks. The natnet method, `addlistener`, needs two inputs. The first is the listener slot to use and the second is the name of the function file to be attached to the listener. This function will execute each time a frame comes from the host.

The listener is turned off by default once it is created. This is to ensure the user has control of the execution of the event callback function. Start the stream with the `enable` method.

```
Command Window
fx >> c.enable(0)
```

The input here represent the index of the listener to enable. So to enable the 5th listener in the list, enter 5. Entering a value of 0 will enable all listeners.

There are 3 callback functions that ship with the natnet class. If they are added to the natnet listener list and enabled, they will execute each time the host sends a frame of data. The `setup.m` file, contains an example of how to operate the class. To stop streaming, use the `disable` method and be sure to enter a value of 0 to disable all listeners.

```
Command Window
fx >> c.disable(0)
```

### IV. Motive control

The natnet class also has functionality to control the Motive application. To enable recording use the startRecord and stopRecord methods, and for playback use the startPlayback and stopPlayback methods. There are a number of additional commands as shown below.

Methods for class natnet:

addlistener	getFrameRate	setPlaybackTakeName
connect	getModelDescription	setTakeName
cycleRecord	liveMode	startPlayback
delete	natnet	startRecord
disable	setAssemblyPath	stopPlayback
disconnect	setPlaybackCurrentFrame	stopRecord
editMode	setPlaybackEndFrame	verifyConnection
enable	setPlaybackLooping	
getFrame	setPlaybackStartFrame	

To display the actions of the class, set the IsReporting property to true. This displays operations of the class to the Command Window.